Requirements for the Information and Monitoring Service

Brian Tierney, Jennifer Schopf

(much of this text is borrowed from the Replication Services Requirements document, Mike Wilde, Koen Holtman, as many of the requirements are the same)

*** DRAFT ***
Feb 12 9 2002

Introduction

This document gives requirements for the Information and Monitoring Service that is an element of Grid architectures of the GriPhyN, PPDG, and EU DataGrid projects.

Information and Monitoring Services are combined into this single document because there are many common requirements. We define these are follows:

- Information Services: contains the state of "static" information
- Monitoring Services: contains the state of "dynamic" information

Static information includes monitoring events such as CPU speed or OS version, where dynamic information includes events such as CPU load of application trace data. There are some fundamental differences in requirements for each type of system. For example, information services only need query/response semantics, but monitoring systems also require publish/subscribe semantics. In general, requirements for information systems are a subset of requirements for monitoring, so the requirements below are actually for monitoring systems.

In this document we use the GMA terminology of producer, consumer, and directory service.

- producer: any component that publishes monitoring data
- consumer: any component the requests monitoring data from a producer
- directory service: a database containing information on what producer publishes what events, and what are the event schemas for those events

An information and monitoring service may be implemented as a single server, as two separate services, or as two services that share the same directory server. The document does not specify an implementation strategy, only requirements. An analysis of existing systems and weather or not they fulfill these requirements needs to be performed.

Architectural and usage assumptions

Architectural requirements specifications of the Information and Monitoring Service are found in several documents:

- the EU DataGrid and WP3 architecture document
- the GriPhyN/PPDG Data Grid Reference Architecture
- the CMS Data Grid system overview and requirements document

We believe that all of these requirements statements are mutually compatible, even though there are differences in terminology, and even though some require more features than others.

Requirements of information and / or monitoring services requirements are also covered in the following documents or web pages:

- MDS Paper (http://www.globus.org/research/papers/MDS-HPDC.pdf)
- GGF Monitoring Documents
  - http://www-didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-3-1.pdf
  - http://www-didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-9-2.pdf
- EDG WP3 Architecture Paper (not on line?)
- http://www-unix.mcs.anl.gov/~schopf/pg-monitoring/

Functionality Requirements

The following is a (random) list of functionality required by an information and monitoring system.

- Monitoring Sensors must be unobtrusive: An exact definition of "unobtrusive is difficult to determine, but this is generally been defined as using less than 1% of the resource running the sensor.

- Local information must be local. To find data on local resources, one must be able to query a local service for it.

- All events must use common timestamps: Recommend use of IETF format: http://www.ietf.org/internet-drafts/draft-ietf-impp-datetime-05.txt

- Soft State Updates must be supported (see MDS paper for explanation of this requirement)

- Access methods: Users and agents need to access sensor data both as a last value and as a continuous stream of data. In other words, both query / response and publish / subscribe semantics must be supported.

- Events must be self-describing: There must be enough information contained with a monitoring event to interpret it (e.g.: units, timestamp, clock accuracy, measurement accuracy, etc) (more on this!)

- Security – currently no use case has security requirements, but there needs to be a security model defined

- Archiving requirements:
    - Length of time data is stored: in general about 6 months, but depends on usage. E.g.: longer if used for accounting
    - Size of data store – dependent on length of time, and size of data, but potentially several GBs.

- Event Schema – what kind of requirements are there for this? Probably GOS satisfies this requirement.

- Dependencies: Several higher-level services have been recognized as being needed for a fully functional end-to-end monitoring and discovery service to be useful these are
    - Archiving
    - Predictive frameworks
    - Alarms/alerts

Performance Requirements for Directory Service:

The numeric requirements in this document are written under the following assumptions.

- Each virtual organization operates a single Information and Monitoring service inside its data grid system
- The service is not necessarily implemented with a single server machine, in fact in the longer term it is more likely that several machines will be used as servers and/or caches.
- From everywhere in the virtual organization's grid, an application executable can use the Information and Monitoring Service by calling functions provided by a library linked in with the executable.

The discovery service has the following requirements, detailed below:

- Capacity
- Frequency of registering sensors
- Frequency of queries on info service
- Consistency – how long should it take for a newly registered sensor to be recognized by all the information service access points?

Capacity

Number of unique event identifiers in the directory service:

(these numbers are all just guesses)

This is based on:

Number of Compute Elements: 10000
Number of events / compute element: 25
Number of Storage Elements: 100
Number of Events / storage elements: 25
Number of network monitoring events, which is based on:
Number of event types: 20

Number of source /destination pairs being monitored: 1000
Number of application events: 10000

Total: about 300,000

Number of users (processes simultaneously running that use the service):
2001: 1000
2006: 10000

(interpretation of the above numbers: if the service were implemented so that every user would have a continuous tcp/ip connection to it, there would be 1000 connections into the service in 2001.)

Access rates

Number of update operations completed per minute. An update can be the insertion of new producer into a directory service.
2001: NN/minute
2006: NNN/minute

Number of lookup operations completed per minute. These numbers are taken to be 'simple' lookup operations that locate information associated with a single producer.
2001: 60,000/minute?
2006: 600,000/minute?

Note: 60,000 operations per minute does not mean that a single operation must be completed in 1/60,000 minute = 1/1000th second, as operations may be processed concurrently.

FOR COMPARISON:
- www.tpc.org has SQL database transaction processing benchmark results. Highest result on benchmark TPC-C is for a configuration with 36 server machines: 709,000/minute. For the Dell Poweredge 4400 (current frontend machine of Caltech tier2 farm, 2-CPU machine): 16,263/minute.

- www.tpc.org has TPC-W WIPS (web interactions per second for a web shopping workload) results. These counts are in completed HTTP requests per second for a web shopping workload. Highest result: 7,073/second (424,380/minute) for a service configuration with 23 server machines.

END OF COMPARISON

Performance Requirements for Producers:

A producer needs to be able to support roughly 10? "subscription" consumers and 10? "query" consumers at any time. This depends on what event data is being published by this producer.

Relaxed consistency metrics

Delay time allowed (in seconds) for an update to become visible to lookup operations performed anywhere in the grid

30 seconds?

Delay time allowed for an update to become visible to lookup operations performed on the same grid site that invoked the update operation

5 seconds?

Service crash recovery requirements

We know of no numeric requirements for service crash recovery. One general assumption is server daemons involved in providing the service should be able to restart automatically with no manual intervention needed if the system they run on crashes and is rebooted. If the contents of the catalog become corrupted or lost through hardware or software failure, there are different strategies for restoring it.

Availability requirements in the case of network or hardware failure

There is no hard numeric requirement we know of for, say, 99.N% availability, or continued service if less than X simultaneous component failures, etc.

Rather the general requirement is that the service makes a good effort of being tolerant of network and hardware failures. How much of an effort is required is very much a function of what the failure characteristics of real-life systems and networks are. This is an area that needs more study. (e.g. preliminary results based on PingER data indicate that network connectivity between CERN and the average site in western Europe and the US can reach CERN is only down for about 0.5-2% of the time.)

Related Documents / URLs:

http://www-didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-16-2.pdf
http://marianne.in2p3.fr/datagrid/documentation/MDS-Deployment-PM9-WP3.doc
http://www.globus.org/gt2/mds2.1/
http://marianne.in2p3.fr/datagrid/documentation/rgma-guide.pdf
http://marianne.in2p3.fr/datagrid/testbed1/documentation/grm-manual.txt
http://www-unix.mcs.anl.gov/gridforum/gis/reports/gos-v3/gis-wg-021-002.html
http://www-unix.mcs.anl.gov/~schopf/pg-monitoring/